

耦合辅助信息的矩阵分解推荐模型^{*}

蒋伟, 秦志光

(电子科技大学 信息与软件工程学院, 成都 610054)

摘要: 近十年来, 协同过滤 (collaborative filtering, CF) 推荐系统成功地为用户提供个性化的产品和服务。然而, 用户-物品矩阵的稀疏性、推荐精度不高等问题仍然是一个挑战。针对这些问题, 在矩阵分解模型基础上, 提出了耦合用户和物品辅助信息的矩阵分解混合协同过滤框架; 然后, 基于此框架又提出了耦合物品属性信息 COS (coupled object similarity) 相似度的过滤模型。大规模真实数据集上的实验表明, 该模型不但可以有效解决物品相似度度量问题, 而且相比传统方法, 尤其是在物品特征非常稀疏的情况下, 推荐准确性得到有效改进。

关键词: 推荐系统; 混合协同过滤; 矩阵分解; 物品相似度; 耦合对象相似度; 辅助信息

中图分类号: TP391 doi: 10.3969/j.issn.1001-3695.2018.03.0206

Matrix factorization recommendation model based on side information

Jiang Wei, Qin Zhiguang

(School of Information & Software Engineering, University of Electronic Science & Technology of China, Chengdu 610054, China)

Abstract: Collaborative filtering (CF) recommender systems have been used to provide users with personalized products and services successfully in the past decade. However, sparseness of user-item matrix and the low accuracy still remain a challenge. To solve these problems, this paper proposed an ensemble framework based on matrix factorization CF for integrating side information of users and items. Based on this framework, this paper proposed a hybrid CF model for integrating COS (Coupled Object Similarity) of attribute information of items. Extensive experiments conduct over large-scale real-word datasets demonstrate that the proposed approach can effectively solve the problem of item similarity measurement, and compared with the traditional approaches, especially in the case of very sparse feature, the accuracy of the recommendation is improved effectively.

Key words: recommender system; hybrid collaborative filtering; matrix factorization; item similarity; coupled object similarity; side information

0 引言

推荐系统作为一种重要的信息过滤工具, 其目标是解决信息过载问题, 并为用户提供个性化的产品和服务, 或者帮助用户决策。相关推荐技术在信息检索、机器学习以及数据挖掘中得到了广泛的研究。在解决信息过载领域, 除搜索引擎外, 推荐系统在互联网上发挥着最为最重要的角色, 目前推荐系统已在 Amazon、Netflix、Youtube 等在线系统成功部署。

个性化推荐方法可以分为三大类: 基于内容的推荐、协同过滤方法、以及结合这两者的混合推荐技术。基于内容的推荐方法^[1,2]从一组被某用户评过分的物品中识别共同特征, 然后向该用户推荐具有这些特性的新物品。这种推荐系统普遍存在两

个问题^[3]: a) 有限内容分析 (limited content analysis), 即物品的特征抽取一般很困难;b) 过度专业化 (over-specialization), 即它无法挖掘出用户的潜在兴趣, 这种推荐系统只依赖于用户过去对某些物品的喜爱, 所以它产生的推荐也都会和用户过去喜欢的物品相似, 无法给用户推荐新的内容。因此, Google, Netflix 等许多公司, 都把注意力转移到协同过滤系统上。

协同过滤方法不依赖于内容信息, 它是目前最受欢迎且最广泛使用的推荐方法。其中, 以隐因子模型为代表的协同过滤方法获得了学术界和产业界的极大关注。它通过发现用户或物品之间的潜在关联来计算推荐, 它把用户和物品都转换到一个相同的隐语义空间^[4,5], 这个隐语义空间同时刻画用户和物品, 并且试图通过这些隐语义来解释用户对物品的偏好程度, 其中

收稿日期: 2018-03-27; 修回日期: 2018-05-11 基金项目: 四川省科技计划资助项目 (2015JY0178, 2014GZ0109, 2015KZ002, 2015JY0030); 国家自然科学基金资助项目 (61472064); 中央高校基本科研基金资助项目 (ZYGX2014J051, ZYGX2014J066)

作者简介: 蒋伟 (1976-), 男, 四川绵阳人, 博士研究生, 主要研究方向为数据挖掘、机器学习 (wayne_jiang@sina.com); 秦志光 (1956-), 男, 教授, 博士, 主要研究方向为计算机开放系统与安全。

评分就是用户对物品偏好程度的一种体现。协同过滤方法克服了一些基于内容方法的局限性。例如: 当得不到内容信息时, 协同过滤仍然可以根据其他用户的反馈来作推荐; 因为协同过滤不依赖于物品内容, 只要有其他用户对相关物品显示出兴趣偏好, 这些物品即使具有不同的内容, 也能把他们推荐给用户。但协同过滤方法也有其固有的问题, 最为典型的是稀疏性及冷启动问题^[6]。例如, 当用户对物品评分数很少时, 很难推断用户的偏好, 从而无法为其做出准确的推荐。

随着 Web 2.0 网站及应用的快速发展, 用户或物品的辅助信息, 例如物品属性信息, 用户画像信息、社交及评论等信息比较容易获取。而这些信息在一定程度上可以体现用户、物品的特征或者用户的偏好。因此, 如何在传统隐因子协同过滤方法中集成这些辅助信息形成混合协同过滤推荐, 使得模糊和不全面的隐语义更加明确, 并最终提高推荐的精度, 缓解冷启动及稀疏性问题, 是近年来需要解决的急迫问题。基于此, 本文基于传统的矩阵分解^[7]隐因子模型, 提出一种耦合用户及物品辅助信息的混合协同过滤框架, 采用用户和物品内容相关联的语义来修正用户的偏好, 从而把用户和物品内容相关的语义集成到矩阵分解中。最后, 基于此框架, 我们提出了一种耦合物品属性相似度的混合协同过滤模型, 来改善矩阵分解协同过滤的推荐性能及冷启动问题。

1 矩阵分解混合协同过滤现状

目前, 协同过滤方法中最流行的评分预测技术是矩阵分解。矩阵分解起源于 2007 年 Netflix 竞赛, 从那时开始, 它便引起了工业界和学术界的大量关注, 并且已经开发了很多矩阵分解的变种。其中一些人利用辅助信息提高用户评分的预测精度。可以融入矩阵分解的辅助信息多种多样, 其中隐式反馈、物品元数据(例如物品属性)、标签、社交信任信息是最为常见的几种。

当显式反馈无法获取, 或者不能完整获取时, 隐式反馈是获取用户偏好的主要方式^[8,9]。Hu 等人^[10]发现隐式数据可以视为积极或消极偏好的标志, 并且提出了偏好程度的计算方法, 这就使得隐式反馈可以作为训练数据。之后, Koren^[5]首先把矩阵分解模型与邻域模型进行混合, 提高了预测准确性。更近一步, 他又提出了支持隐式反馈的矩阵分解模型 SVD++^[7]。SVD++将用户提供的隐式反馈如用户的浏览历史, 或用户购买的物品信息以隐因子的方式融入矩阵分解模型中, 取得了很好的效果, 并最终成为 Netflix 竞赛的获胜方案。

关于元数据的集成, 很多基于内容的方法^[11]中都有论述。这种方法设计出物品属性和用户画像之间的匹配机制, 从而使推荐行为类似于信息检索系统。然而, 基于内容的过滤算法通常有其相关的问题, 如冷启动和过于专业化。基于矩阵分解的协同过滤可以一定程度上缓解这些问题。它的主要思想是计算用户或物品的相似性, 以预测新物品给用户, 并利用矩阵分解来减少用户-物品矩阵的维度^[5,7]。manzato^[12] 创建了一个用户

-类别矩阵分解模型来提取用户对电影的偏好, 并利用矩阵分解技术来计算用户相似度, 利用相似度获得用户的偏好类别, 最后用协同过滤方法进行推荐。最近此作者对 SVD++进行了扩展, 提出了 gSVD++^[13]模型。在 gSVD++中, 除了用户的隐式反馈, 元数据的隐式反馈也融入到矩阵分解模型中。Gantner 等人^[14]将用户或物品属性相关的元数据也映射到矩阵分解模型中, 通过这种映射, 使得这种模型既能通过标准技术进行训练, 也可以解决用户和物品的冷启动问题。

Enrich 等人^[15]通过引入标签信息来改进 SVD++算法。在冷启动的情况下, 取得了更高的精度。与 Enrich 等人类似, Fernández-Tobías 等人^[16]基于 gSVD 提出了 tagGSVD 模型, 把用户标签和物品标签集成到了矩阵分解模型中, 该模型在跨域推荐中提高了系统性能。而文献[17]基于 gSVD 方法在食品推荐中集成了社会标签。

基于信任感知的矩阵分解方法, 按信任信息与矩阵分解模型的集成方式上来说, 具有代表性的有如下的几个模型: SoRec^[18]在优化函数层面把用户-物品评分矩阵分解和用户-用户社交信任矩阵分解线性叠加在一起, 这两个矩阵分解共享相同的用户隐式空间。STE^[19](Social trust ensemble, STE)方法在评分预测函数上把评分矩阵分解方法和社交矩阵分解线性组合在一起, 得到用户对物品的评分。这种方法不仅考虑了用户自身的偏好对于评分的影响, 而且考虑了用户的朋友对其评分的影响。SocialMF^[20]通过在矩阵分解模型的优化函数中添加社会正则约束项来实现用户隐特征向其朋友隐特征逼近, 并且解决了社会网络中的信任传播问题。该文献认为, 对于 RMSE 指标而言, 其推荐效果要好于 SoRec 和 STE 模型。文献[21]基于 SocialMF 提出了另一种社会正则化约束项。不同于[20]中的基于平均的正则化项, 该文献采用的是基于个体的正则化项。

如果每个物品具有很多属性或者辅助信息, 而且维度可能很高或者信息相当冗余, 这时传统的矩阵分解隐因子分解模型很难捕捉非线性特征表示^[22]。AutoSVD++^[23]采用压缩自编码器学习物品特征表示, 然后集成这些特征到矩阵分解模型中。

本文基于矩阵分解协同过滤, 提出了耦合用户和物品辅助信息的混合协同过滤模型框架, 以及耦合物品相似度的模型算法。论文的论述流程如图 1 所示, 其主要贡献是:

- a) 基于矩阵分解协同过滤模型, 提出了融合用户或物品辅助信息的混合过滤模型框架。
- b) 在模型框架基础上, 提出了融合物品属性相似度的矩阵分解混合协同过滤方法; 并且在模型中引入了耦合对象相似度(coupled object similarity, COS)来计算物品属性之间的相似性。
- c) 大规模的真实用户数据上的实验表明, 本文提出的基于物品属性相似度的矩阵分解混合协同过滤模型可以显著提高评分预测准确性, 并且可以缓解冷启动和稀疏性问题。

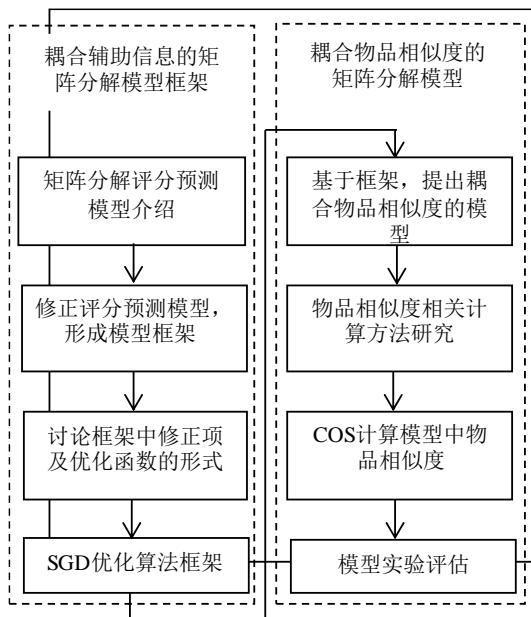


图 1 论述流程

2 矩阵分解协同过滤

2.1 基线预测器

基线预测器（也可称为基线估计器）根据用户或物品自身内在的特点，从训练数据捕捉系统性偏差（bias）。用户对物品的评分就可能存在这种系统性偏差^[24]。例如，假设所有物品的平均分为 3 分，因为某物品本身的品质等因素，它比平均分要高 0.8 分；另一方面，由于某用户比较挑剔，他对此物品的评分倾向于比平均分低 0.3 分；在这种情况下，此用户对这个物品的评分估计值为 $3+0.8-0.3=3.5$ 。

针对评分预测的情形，用户集合表示为 $U = \{u_1, u_2, \dots, u_m\}$ ，物品集合表示为 $I = \{i_1, i_2, \dots, i_n\}$ ，用户对物品的评分用矩阵表示为 $R = [r_{u,i}]_{m \times n}$ ，其元素 $r_{u,i}$ 表示用户 $u \in U$ 对物品 $i \in I$ 的评分。通常 $r_{u,i} \in [1, 5]$ ，且 $r_{u,i}$ 可为整数，也可以为实数。不失去一般性，我们把 $r_{u,i}$ 通过正则化方法映射到区间 $[0, 1]$ 。

基线预测器可以通过调整数据从而克服偏差问题。对于用户 u 对物品 i 的真实评分 r_{ui} 的基线预测器可以表示为

$$b_{ui} = \mu + b_u + b_i$$

这里参数 μ , b_u , b_i 分别表示平均值，用户 u 的偏差和物品 i 的偏差。有多种方法可以估计 b_u 和 b_i ，下面是常用的两种方法。

a) 简单粗糙方式。首先从 $b_u + b_i (= b_{ui} - \mu)$ 中估计出 b_i ：

$$b_i = (\sum_{u \in R(i)} (r_{ui} - \mu)) / (\lambda_1 + |R(i)|)$$

然后，根据 b_i 可以计算出 b_u ：

$$b_u = (\sum_{i \in R(u)} (r_{ui} - \mu - b_i)) / (\lambda_2 + |R(u)|)$$

其中 $R(i)$ 、 $R(u)$ 分别是对 i 评过分的用户集合，以及被用户 u 评分的物品集合。 λ_1 、 λ_2 是正则化参数，通过交叉验证决定，

例如对于 Netflix 数据集他们分别是 $\lambda_1=25$ 、 $\lambda_2=10$ 。

b) 复杂但更为准确的方法。通过解下列最小平方误差问题而求得：

$$\min_{b_u, b_i} \sum_{(u, i) \in R} (r_{ui} - \mu - b_u - b_i)^2 + \lambda (\sum_u b_u^2 + \sum_i b_i^2)$$

式子中第一项试图寻找特定的用户和项目的偏差 b_u 、 b_i 来适应给定的评分数据；第二项用于防止过拟合。

2.2 过滤模型

矩阵分解协同过滤方法借用了隐语义分析技术^[25]，这种技术通过词汇-文档矩阵低秩奇异值分解来自动推断隐式概念。推荐系统的矩阵分解在用户-物品矩阵基础上完成评分预测任务，但它只是针对观测到的评分矩阵而不是整个矩阵。

矩阵分解模型将用户和物品映射到低维（设维度为 K ）的联合隐特征空间，而用户-物品交互（如评分信息）可以被建模为该空间中的内积。每个用户 u 对应于一个列向量 $P_u \in \mathbb{R}^K$ ，每个物品 i 对应于一个行向量 $Q_i \in \mathbb{R}^K$ 。对于一个给定的用户 u ， P_u 的元素度量了这个用户对相应的物品特征感兴趣的程度。对于给定的物品 i ， Q_i 的元素度量此物品拥有这些特征的程度。 m 个用户和 n 个物品分别形成用户隐特矩阵 $P \in \mathbb{R}^{m \times K}$ 和物品隐特征矩阵 $Q \in \mathbb{R}^{n \times K}$ 。内积 $P_u^T Q_i$ 就是用户 u 对物品 i 交互的建模。所以，若给定特征向量维数 K ，用户-物品评分矩阵可以分解为 P 和 Q 两部分：

$$R \approx \hat{R} = P^T Q \text{ 或写作 } \hat{r}_{u,i} = P_u^T Q_i$$

再结合基线估计器，从而模型变为

$$\hat{r}_{u,i} = \mu + b_u + b_i + P_u^T Q_i$$

通过使用观测到的评分数据，最小化如下的目标函数来学习隐特征矩阵 P 和 Q ，即

$$L = \frac{1}{2} \sum_{(u, i) \in R} (r_{ui} - \hat{r}_{u,i})^2 + \frac{\lambda}{2} (b_u^2 + b_i^2 + \|P_u\|_F^2 + \|Q_i\|_F^2)$$

R 为可观测用户-物品对的集合， $\|\cdot\|_F^2$ 为 Frobenius 范式，

正则化 $\frac{\lambda}{2} (b_u^2 + b_i^2 + \|P_u\|^2 + \|Q_i\|^2)$ 用来避免过拟合。 λ 为正则化超参数，用来控制正则化项对隐特征向量的影响。我们通常使用随机梯度下降法（stochastic gradient descent, SGD）求解此目标函数的局部最优解。

实际上，可以针对用户偏差、物品偏差、用户隐因子、物品隐因子分别设置不同的正则化超参数 λ 和训练学习率 η ，以提高预测准确性。相关策略的深度讨论可以参考文献[26]。

3 矩阵分解混合协同过滤模型框架

3.1 模型框架的提出

从隐语义上理解，矩阵分解模型把用户对物品的偏好程度表示为评分 \hat{r} ，其元素 $\hat{r}_{u,i} = P_u^T Q_i$ ，采用梯度下降迭代算法把偏好程度 \hat{r} 分解成用户的口味 P_u ，以及与用户口味对应的物品特征 Q_i ，但用户的口味 P_u 和物品特征 Q_i 是通过已经观测到的评分数据得来，因此它们只是真实特征的近似。另外，可以根

据物品和用户的其他内容、属性、社交等信息分别得到用户口味增量 ΔP_u 以及项目特征增量 ΔQ_i , 可以据此修正用户的口味以及物品的特征来更好地逼近用户和项目的真实特征, 也就是在算法迭代过程中不断采用 $P_u = P_u + \alpha \Delta P_u$, $Q_i = Q_i + \beta \Delta Q_i$ 对用户的口味 P_u 和项目特征 Q_i 进行更新, 其中 $\alpha, \beta \in [0, 1]$ 分别是 P_u 和 ΔP_u 之间的权重参数, 以及 Q_i 和 ΔQ_i 之间的权重参数。因此, 模型框架可以描述如下:

$$\hat{r}_{u,i} = \mu + b_u + b_i + (P_u + \alpha \Delta P_u)^T (Q_i + \beta \Delta Q_i)$$

$$L = \frac{1}{2} \sum_{(u,i) \in R} (r_{ui} - \hat{r}_{ui})^2 + \frac{1}{2} f_{reg} + \frac{1}{2} f_{reg \Delta P} + \frac{1}{2} f_{reg \Delta Q}$$

$$f_{reg} = \lambda_1 b_u^2 + \lambda_2 b_i^2 + \lambda_3 ||P_u||^2 + \lambda_4 ||Q_i||^2$$

$f_{reg \Delta P}$ 是防止 ΔP_u 过拟合的正则化项

$f_{reg \Delta Q}$ 是防止 ΔQ_i 过拟合的正则化项

为了灵活起见, 框架中每一个正则化项采用不同的超参数变量。 α, β 为叠加权重系数。

如果 $\alpha \Delta P_u = 0$ 或者 $\beta \Delta Q_i = 0$, 则模型忽略了相应的修正, 其行为与标准矩阵分解一样。

当 P_u, Q_i 有多个修正项目时可以在模型中连续线性叠加, 构成如下的预测模型:

$$\begin{aligned} \hat{r}_{u,i} &= \mu + b_u + b_i + (P_u + \alpha_1 \Delta P_{u1} + \alpha_2 \Delta P_{u2} + \dots)^T \times \\ &\quad (Q_i + \beta_1 \Delta Q_{i1} + \beta_2 \Delta Q_{i2} + \dots) \end{aligned}$$

为了简单起见, 接下来的讨论假设 P_u, Q_i 分别最多只有一个修正项 $\alpha \Delta P_u, \beta \Delta Q_i$ 。

3.2 $\alpha \Delta P_u$ 和 $\beta \Delta Q_i$ 的讨论

1) $\alpha \Delta P_u$ 的讨论

a) ΔP_u 是只含有一个隐特征向量 x_v 的函数, 且该函数与 P_u

无关, 那么, (a) $f_{reg \Delta P} = \lambda_p \sum_{v \in N(u)} ||x_v||^2 // N(u)$ 为用户 u 相关的集合, 其元素可能为用户、物品或他们之间交互的内容; (b) 如果 ΔP_u 是 x_v ($v \in N(u)$) 的线性组合, 则模型框架中 $\alpha = 1 //$ 从而可以直接去掉 α , 其证明如图2所示。

否则 $f_{reg \Delta P} = 0$ 。

由已知条件得: $\alpha \Delta P_u(x_v) = \alpha \sum_{v \in N(u)} n_v x_v = \sum_{v \in N(u)} n_v (\alpha x_v)$ $f_{reg \Delta P} = \lambda_p \sum_{v \in N(u)} x_v ^2 = (\lambda_p / \alpha^2) \sum_{v \in N(u)} \alpha x_v ^2$ $\Leftrightarrow \dot{x}_v = \alpha x_v, \dot{\lambda}_p = \lambda_p / \alpha^2 \text{ 则有}$ $\alpha \Delta P_u(x_v) = \Delta P_u(\dot{x}_v)$ $(\lambda_p / \alpha^2) \sum_{v \in N(u)} \alpha x_v ^2 = \dot{\lambda}_p \sum_{v \in N(u)} \dot{x}_v ^2$ $\text{最后再把 } \dot{x}_v, \dot{\lambda}_p \text{ 变量名称换回 } x_v, \lambda_p \text{ 则:}$ $\hat{r}_{u,i} = \mu + b_u + b_i + (P_u + \Delta P_u)^T (Q_i + \beta \Delta Q_i) \quad (\alpha \text{ 被去掉了})$ $L = \frac{1}{2} \sum_{(u,i) \in R} (r_{ui} - \hat{r}_{ui})^2 + \frac{1}{2} f_{reg} + \frac{1}{2} f_{reg \Delta P} + \frac{1}{2} f_{reg \Delta Q}$ $f_{reg \Delta P} = \lambda_p \sum_{v \in N(u)} x_v ^2$ <p>在上述所有推导过程中都是对中间变量作线性变量代换, 所以变换前后两个模型 $\hat{r}_{u,i}$ 与 L 在预测评分这个问题上实际上是等价的。</p>
--

图 2 1)(1)(b) 结论的证明

b) ΔP_u 只是隐特征向量 $P_k (k \in U)$ 的函数时,

$f_{reg \Delta P} = 0$ //因 f_{reg} 中含有 $\lambda_p ||P_u||^2$, 故无须再添加防止过拟合的正则化项, 即 $f_{reg \Delta P} = 0$ 。

2) $\beta \Delta Q_i$ 的讨论

a) ΔQ_i 是只含有一个隐特征向量 y_j 的函数, 且该函数与 Q_i

无关, 那么, (a) $f_{reg \Delta Q} = \lambda_y \sum_{j \in N(i)} ||y_j||^2 // N(i)$ 为物品 i 相关的集合, 其元素可能为用户、物品或者他们之间交互的内容; (b) 如果 ΔQ_i 是 $y_j (j \in N(i))$ 的线性组合, 则模型框架中 $\beta = 1 //$ 从而可以直接去掉 β , 其证明与1)中(b)结论的证明类似。

否则: $f_{reg \Delta Q} = 0$

b) ΔQ_i 只是隐特征向量 $Q_k (k \in I)$ 的函数时,

$f_{reg \Delta Q} = 0$ //因 f_{reg} 中含有 $\lambda_q ||Q_i||^2$, 故无须再添加防止过拟合的正则化项, 即 $f_{reg \Delta Q} = 0$ 。

3.3 模型优化算法

1) SGD 算法

通过对损失函数的最小化, 模型的参数可以从观测到的训练数据中自动学习。我们可以使用梯度下降算法来求函数 L 的局部最小值。具体方法是, 针对每一对观测到的训练数据 $(u, i) \in R$, 迭代更新模型参数。假设 θ 是一个模型参数, 随机梯度下降法让 θ 在损失函数 L 的局部损失下降最快的方向进行移动, 从而求出局部最小值, 每一步 θ 值由它的梯度来表示:

$$\theta \leftarrow \theta - \eta \frac{\partial L}{\partial \theta}$$

这里 η 是学习率, 它决定参数在相关梯度方向上的更新程度。

学习率小则学习慢, 然而如果学习率太大则算法不能收敛。

在模型框架中, 损失函数关于各个参数的导数如下所示:

//设 $e_{ui} = r_{ui} - \hat{r}_{ui}$ 是评分 r_{ui} 的预测错误值。

$$\frac{\partial L}{\partial b_u} = -e_{ui} + \lambda_1 b_u$$

$$\frac{\partial L}{\partial b_i} = -e_{ui} + \lambda_2 b_i$$

$$\frac{\partial L}{\partial P_u} = -e_{ui} (Q_i + \beta \Delta Q_i) \left(1 + \alpha \frac{\partial \Delta P_u}{\partial P_u} \right) + \lambda_3 P_u$$

//如果 ΔP_u 与 P_u 无关, 则 $\frac{\partial \Delta P_u}{\partial P_u} = 0$

$$\frac{\partial L}{\partial Q_i} = -e_{ui} (P_u + \alpha \Delta P_u) \left(1 + \beta \frac{\partial \Delta Q_i}{\partial Q_i} \right) + \lambda_4 Q_i$$

//如果 ΔQ_i 与 Q_i 无关, 则 $\frac{\partial \Delta Q_i}{\partial Q_i} = 0$

//如果存在隐特征 x_v 我们可以按如下方式计算 $\frac{\partial L}{\partial x_v}$:

$$\frac{\partial L}{\partial x_v} = -\alpha \frac{\partial \Delta P_u}{\partial x_v} e_{ui} (Q_i + \beta \Delta Q_i) + \lambda_p x_v$$

//如果存在隐特征 y_j , 可以按如下方式计算 $\frac{\partial L}{\partial y_j}$:

$$\frac{\partial L}{\partial y_j} = -\beta \frac{\partial \Delta Q_i}{\partial y_j} e_{ui} (P_u + \alpha \Delta P_u) + \lambda_q y_j$$

2) 优化算法

根据以上的导数, 可以设计如下所示的随机梯度下降算法。

SGD算法 (Stochastic Gradient Descent)

procedure TRAIN

 Initialize the parameters at random

 for epoch $\leftarrow 1, \dots, N$ or $|L^{epoch} - L^{epoch+1}| < \varepsilon$ do

 SHUFFLE(R)

 for all $(u, i) \in R$ do

$$b_u \leftarrow b_u - \eta \frac{\partial L}{\partial b_u}$$

$$b_i \leftarrow b_i - \eta \frac{\partial L}{\partial b_i}$$

$$P_u \leftarrow P_u - \eta \frac{\partial L}{\partial P_u}$$

$$Q_i \leftarrow Q_i - \eta \frac{\partial L}{\partial Q_i}$$

 //if latent factor feature x_v exists then we can
 compute x_v as follows:

 for all $v \in N(u)$ do

$$x_v \leftarrow x_v - \eta \frac{\partial L}{\partial x_v}$$

 end for

 //if latent factor feature y_j exists then we can
 compute y_j as follows:

 for all $j \in N(i)$ do

$$y_j \leftarrow y_j - \eta \frac{\partial L}{\partial y_j}$$

 end for

 end for

end for

end procedure

在离线环境中, 这个过程通常在训练集上重复固定次数 N 或者两次损失误差函数差值很小如 $\varepsilon = 0.0001$ 时停止。

模型参数 b_u 、 b_i 的初始化值, 可根据基线预测器的方法获得, 而参数 P 、 Q 可以通过深度自编码器进行预训练得到。另外这些参数也可以用随机方法进行初始化。

3.4 模型的应用

现有融合辅助信息的矩阵分解混合协同过滤模型中, 有许多符合我们的模型框架思想。例如, Koren等人的SVD++^[7]采用用户与物品的交互信息 (评分、浏览、购买等) 来修正用户隐

特征向量; Manzat^[14]基于SVD++提出了gSVD++, 他用物品相关的元数据信息, 例如电影的风格、演员阵容、发行商和关键词等, 来修正物品的隐特征向量; AutoSVD++^[23]使用压缩自编码器从物品的辅助信息提取特征表示, 并且用于修正物品的隐特征向量; Fernández-Tobías等人^[17]采用用户与物品交互的标签信息分别捕捉用户的偏好信息与物品的特征信息, 并分别把他们用于修正用户及物品的隐特征向量。

接下来基于这里的模型框架, 提出一种融合物品信息相似度的混合过滤模型, 并通过实验来验证其推荐性能。

4 耦合物品相似度

4.1 耦合模型

为了把物品属性集成到本文的模型, 采用物品属性相似度信息来修正物品的特征, 以提高推荐准确性、减轻项目端的稀疏性及冷启动问题。基于提出的模型框架, 可以按如下方法构建模型:

1) 与修正用户偏好相关的成分: 由于没有集成反映用户偏好的数据, 故在模型中与修正用户偏好相关的成分均为0。也就是:

$$(1) \quad \alpha \Delta P_u = 0$$

$$(2) \quad f_{reg \Delta P} = 0$$

2) 与修正物品特征相关的成分:

$$(1) \quad \beta \Delta Q_i = \beta \sum_{j \in R} S_{i,j} y_{ij}$$

$$(2) \quad y_{ij} = (Q_j - Q_i)$$

$$(3) \quad f_{reg \Delta Q} = 0$$

其中, $S_{i,j}$ 表示物品 i 和物品 j 之间的相似度。所有物品对之间的相似度构成物品相似度矩阵 S 。相对于评分预测模型来说, ΔQ_i 只跟变量 Q 相关, 没有引入额外的变量。而在损失函数中已包含 Q 的过拟合项, 所以这里不再需要添加防止过拟合的正则化项即 $f_{reg \Delta Q} = 0$ 。

把上面式子代入模型框架得到模型:

$$\hat{r}_{u,i} = \mu + b_u + b_i + (P_u)^T (Q_i + \beta \sum_{j \in R} S_{i,j} (Q_j - Q_i))$$

$$L = \frac{1}{2} \sum_{(u,i) \in R} (r_{ui} - \hat{r}_{u,i})^2 + \frac{1}{2} (\lambda_1 b_u^2 + \lambda_2 b_i^2 + \lambda_3 ||P_u||^2 + \lambda_4 ||Q_i||^2)$$

这个模型把物品隐特征向量分解成 Q_i 与 ΔQ_i 两部分。

ΔQ_i 是 Q_i 的修正部分。另外也可以这样理解: Q_i 表示基于物品的隐偏置向量, 而 ΔQ_i 是从物品属性中提取的特征向量。

4.2 相似度计算

4.2.1 相关相似度

属性相似度用于度量数据对象之间的相似关系的强度, 两个对象越相似, 相似度就越大。对于标量 (也就是无方向意义的数字), 已经存欧氏距离 (Euclidean distance)、曼哈顿距离 (Manhattan distance)、闵可夫斯基距离 (Minkowski distance)

^[27]等各种各样的相似度度量指标。相对而言, 分类变量(二元变量是分类变量的特例)则很少有人关注。而在推荐系统中, 从推荐物品内容中抽取的特征值大多都是分类变量。例如, 在电影推荐系中, 可以抽取电影的 director, actor 和 genre 特征值来描述一部电影, (“Hitchcock”, “Stewart”, “Thriller”) 和 (“Koster”, “Grant”, “Comedy”) 特征向量值分别表示电影 “Vertigo” 和 “Bishop’s Wife”的特征向量。计算类似这些电影数据对象的相似度是基于内容的推荐系统必须解决的核心问题。

HD 距离(Heterogeneous distances)^[28]和 MVDM 距离(modified value distance matrix)^[29], 在监督学习中刻画了分类变量之间的相似度。而在未标定数据中, OF(occurrence frequency)^[30]和 SMS(Simple Matching Similarity)仅仅使用 0 和 1 来区分不同类型值和相同类型值之间的相似度。这些方法都过于粗糙, 不能精确刻画类别属性变量的相似度, 他们没有结合相关方面全面地刻画相似度, 而只刻画了相似度的局部信息, 并且不是数据驱动的方法。Wang 等人^[31~33]提出了适合计算类别型数据描述的物品之间的相似度度量方法: 耦合对象相似度(Coupled Object similarity, COS)。耦合对象相似度同时考虑了属性值在一个属性内耦合相似度和不同属性间的耦合相似度。耦合对象相似度能够以较高的准确度和较低的算法复杂度获得特征值的频率分布情况和特征依赖聚合度。文献[32,33]验证了耦合对象相似度在捕获物品之间的关系上的有效性, 故本文采用耦合对象相似度来度量物品属性之间的相似性。

4.2.2 耦合对象相似度

计算物品相似度的属性信息可以组织成物品信息表的形式, 它是一个二维表, 其中行向量对应各个物品的数据对象, 列向量则分别表示不同属性的值。表1是物品信息表的例子, 它的行向量 $O_1, O_2, O_3, O_4, O_5, O_6$ 分别对应物品 $i_1, i_2, i_3, i_4, i_5, i_6$ 的数据对象, 这些数据对象由相应物品在属性 A_1, A_2, A_3 上的属性值构成。

表1 物品信息表

数据对象	A_1	A_2	A_3
O_1	a_1	b_1	c_1
O_2	a_2	b_1	c_1
O_3	a_2	b_2	c_2
O_4	a_3	b_3	c_2
O_5	a_4	b_3	c_3
O_6	a_4	b_2	c_3

物品 i_1, i_2 之间的耦合对象相似度(Coupled Object Similarity, COS)定义为

$$\text{COS}(O_{i_1}, O_{i_2}) = \sum_{j=1}^n \delta_j^A(x_{i_1,j}, x_{i_2,j})$$

n 为属性个数。 $\delta_j^A(x_{i_1,j}, x_{i_2,j})$ 是物品数据对象 O_{i_1} 在属性 A_j

上的取值 $x_{i_1,j}$, 与物品 O_{i_2} 在属性 A_j 上的取值 $x_{i_2,j}$ 的耦合属性相似度(Coupled Attribute Value Similarity, CAVS)。物品

i_1, i_2 之间的耦合对象相似度为他们在各个属性上的耦合相似度之和。

属性 A_j 的取值 x, y 之间的耦合属性相似度(Coupled Attribute Value Similarity, CAVS)定义为:

$$\delta_j^A(x, y) = \delta_j^{Ia}(x, y) \times \delta_j^{Ie}(x, y)$$

这里 δ_j^{Ia} 和 δ_j^{Ie} 分别是特征内耦合属性值相似度(Intracoupled Attribute Value Similarity, IaAVS)和特征间耦合属性值相似度(Intertribute Attribute Value Similarity, IeAVS)。

在计算属性 A_j 的取值 x, y 之间的特征内耦合属性值相似度(IaAVS)时, 仅考虑同一属性 A_j 内属性值 x 与 y 之间的相互关系, 并且在度量相似度时考虑了同一属性下其取值出现的频率。其定义为

$$\delta_j^{Ia}(x, y) = \frac{|g_j(x)| \times |g_j(y)|}{|g_j(x)| + |g_j(y)| + |g_j(x)| \times |g_j(y)|}$$

式中, $g_j(x)$ 表示物品数据对象集合 O 中, 属性 A_j 上取值等于 x 的数据对象集合, $|g_j(x)|$ 表示这个数据对象集合中集合元素的个数。 $g_j(y)$ 和 $|g_j(y)|$ 与此类似。

属性 A_j 的值 x, y 之间的特征间耦合属性值相似度(IeAVS)考虑了属性 A_j 内取值 x 与 y 之间的特征依赖聚合度。以及在属性 A_j 内取值为 x 与 y 的条件下, 其他属性 A_k ($k \neq j$) 取值的分布情况。其定义为

$$\delta_j^{Ie}(x, y) = \sum_{j=1, k \neq j}^n \alpha_k \delta_{j|k}(x, y)$$

这里 α_k 是属性 A_k 的权重参数, $\sum_{k=1}^n \alpha_k = 1, \alpha_k \in [0, 1]$ 。

$\delta_{j|k}(x, y)$ 是属性值 x 与 y 在特征 A_k ($k \neq j$) 下的特征间耦合属性值相似度, 其定义为

$$\delta_{j|k}(x, y) = \sum_{w \in \cap} \min\{P_{k|j}(\{w\}|x), P_{k|j}(\{w\}|y)\}$$

其中, \cap 为属性 A_j 取值为 x 条件下属性 A_k 的所有取值集合, 与属性 A_j 取值 y 条件下属性 A_k 的取值集合的交集。

$P_{k|j}(\{w\}|x)$ 表示属性 A_j 取值为 x 的条件下, 属性 A_k 取值为 w 的条件概率, 其定义为

$$P_{k|j}(\{w\}|x) = \frac{|g_k(w) \cap g_j(x)|}{|g_j(x)|}$$

$P_{k|j}(\{w\}|y)$ 与此类似。

以表2为例, 计算物品对象 O_4 和 O_5 的耦合相似度:

$$\begin{aligned} \text{COS}(O_4, O_5) &= \delta_1^A(x_{4,1}, x_{5,1}) + \delta_2^A(x_{4,2}, x_{5,2}) + \delta_3^A(x_{4,3}, x_{5,3}) \\ &= \delta_1^A('a_3', 'a_4') + \delta_2^A('b_3', 'b_3') + \delta_3^A('c_2', 'c_3') . \end{aligned}$$

其中 $\delta_1^A('a_3', 'a_4') = \delta_1^{Ia}('a_3', 'a_4') \times \delta_1^{Ie}('a_3', 'a_4')$

由于 O_4, O_5 在属性 A_1 的取值 ' a_3 ', ' a_4 ' 分别在整个属性列 A_1 列出现 1, 2 次, 根据公式可得:

$$\delta_1^{Ia}('a_3', 'a_4') = \frac{1 \times 2}{1 + 2 + 1 \times 2} = 0.4 .$$

$\delta_1^{Ie}('a_3', 'a_4') = \alpha_2 \delta_{1|2}('a_3', 'a_4') + \alpha_3 \delta_{1|3}('a_3', 'a_4')$, 这里属

性的权重参数取相同的值0.5;

$$\delta_{12}^A(a_3, a_4) = \min \{P_{2|1}(b_3 | a_3), P_{2|1}(b_3 | a_4)\}$$

$$= \min \left\{ \frac{|g_2(b_3) \cap g_1(a_3)|}{|g_1(a_3)|}, \frac{|g_2(b_3) \cap g_1(a_4)|}{|g_1(a_4)|} \right\}$$

$$= \min \{1, 1\} = 1;$$

$$\delta_{13}^A(a_3, a_4) = 0 (\text{因为 } \cap = \emptyset);$$

$$\text{故 } \delta_1^{fa}(a_3, a_4) = \delta_1^A(a_3, a_4)$$

$$= \alpha_2 \delta_{12}^A(a_3, a_4) + \alpha_3 \delta_{13}^A(a_3, a_4)$$

$$= 0.5 \times 1 + 0.5 \times 0 = 0.5.$$

$$\text{所以 } \delta_1^A(a_3, a_4) = \delta_1^{fa}(a_3, a_4) \times \delta_1^{fe}(a_3, a_4)$$

$$= 0.4 \times 0.5 = 0.2.$$

采用计算 $\delta_1^A(a_3, a_4)$ 相似的方法, 可以计算出

$\delta_2^A(b_3, b_3)$, $\delta_3^A(c_2, c_3)$ 。他们三项之和就是 θ_4, θ_5 的耦合对象相似度 $\cos(\theta_4, \theta_5)$ 。

5 实验

5.1 实验设置

本实验目的是验证本文提出的耦合物品属性相似度混合协同过滤模型的有效性, 其中的相似度采用耦合对象相似度 (coupled Object similarity) 来计算。

5.1.1 数据集描述

采用公共电影数据集 MovieLens-1M 以及 hetrec2011 来评估提出的模型的有效性。这些数据集由 GroupLens research (<http://www.grouplens.org>) 发布, 其广泛应用于评价协同过滤算法。该数据集除了包括电影评分外, 还有电影风格等元数据。数据集的详细统计数据如表 2 所示。

表 2 统计数据

数据类别	MovieLens-1M	Hetrec2011
用户数量	6040	2113
电影数量	3900	10197
评分数量	1000209	855598
评分数量/电影	256	85
评分数量/用户	165	405
电影风格类型数量	18	20

本实验中, 只考虑电影风格属性。由于同一部电影可能具有多个风格, 把电影风格分解成与其属性类型个数相同的特征, 也就是特征向量 (g_1, g_2, \dots, g_m) , m 为电影风格类型数量, 也即此特征向量的维数。若某电影具有某风格特征则特征向量中对应的 $g_i = 1 (1 \leq i \leq m)$, 否则 $g_i = 0$ 。其中每个特征的权重参数 α_k 设置为 $1 / (m - 1)$ 。

5.1.2 评价指标

采用广泛使用的均方根误差 (root mean squared error, RMSE) 和平均绝对误差 (mean absolute error, MAE) 作为评价指标来度量推荐性能。其中 RMSE 定义为

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (\hat{r}_{u,i} - r_{u,i})^2}$$

MAE 定义为

$$MAE = \frac{1}{|T|} \sum_{(u,i) \in T} |\hat{r}_{u,i} - r_{u,i}|$$

这里 $r_{u,i}$ 是用户 u 对物品 i 的实际评分, $\hat{r}_{u,i}$ 是用户 u 关于物品 i 的评分的预测值。 T 表示测试集中的评分数量。RMSE 和 MAE 越小, 则推荐越准确。

对于所有的实验, 进行5折交叉实验来对提出的算法与其他算法进行比较^[34]。首先将数据集分割成5个大小相同的子集, 其中一个子集被认为是测试数据集, 另外4个子集是训练数据集。重复5次交叉验证的过程, 5个子集中的每一个都在其中一次实验中作为测试集, 剩下的4个子集作为训练集。

5.2 实验结果

5.2.1 不同模型综合性能比较

选择如下已有隐因子分解模型来与我们提出的模型进行比较:

a) MF。矩阵分解模型MF^[35]由Koren等提出, 它仅仅利用用户-物品评分矩阵进行推荐。

b) NMF。NMF^[36,37]即非负矩阵分解, 由Lee and Seung等人最早在[36]中提出, 用于图像分析。它约束用户和物品的隐特征矩阵元素为非负, 它被广泛应用于协同过滤。

c) PMF。Salakhutdinov与Mnih提出的PMF, 即概率矩阵分解^[38], 它使用带高斯噪声的概率模型来表示用户和物品的隐特征向量, 从而实现概率角度的矩阵分解。

d) BPMF。BPMF也是由Salakhutdinov和Mnih在[39]中提出。该模型用贝叶斯方法自动控制的参数和超参数, 并使用马尔可夫链蒙特卡洛方法进行训练。

e) SVD++。SVD++在SVD基础上, 结合邻域模型的优势, 集成了隐式反馈。

对所有模型, 分别设置维数 $K = 10, K = 40$ 进行实验。为了取得公平的比较结果, 根据上述各算法的相关文献设置参数, 以便得到这些模型的最佳性能, 而对于本文的模型, 使用如下的超参数配置: $\eta = 0.01$, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0.1$ 。对于MovieLens-1M和HetRec2011数据集, β 分别设为0.2和0.3。针对每一个模型, 在数据集上进行5次5折交叉验证实验, 取平均的RMSE和MAE作为最终的结果, 其结果如表3和4所示。

表 3 综合性能对比(MovieLens-1M)

K	Metric	MF	NMF	PMF	BPMF	SVD++	Our Model
10	RMSE	0.8776	0.9153	0.8831	0.8970	0.8553	0.8482
	MAE	0.7095	0.7413	0.7143	0.7220	0.7033	0.6918
40	RMSE	0.8895	0.9272	0.8905	0.9042	0.8642	0.8578
	MAE	0.7248	0.7585	0.7290	0.7391	0.7198	0.7065

表 4 综合性能对比(HetRec2011)

K	Metric	MF	NMF	PMF	BPMF	SVD++	Our Model
10	RMSE	0.7915	0.8323	0.8004	0.8041	0.7898	0.7672

40	MAE	0.6151	0.6295	0.6162	0.6193	0.6083	0.5753
	RMSE	0.8241	0.8593	0.8323	0.8357	0.8035	0.7871
	MAE	0.6323	0.6461	0.6354	0.6383	0.6256	0.5956

从实验结果可以看出,除了提出的模型之外, SVD++的推荐性能超过其他所有的方法。这说明隐式反馈信息对提升推荐准确性有很大的帮助。而提出的方法性能要好于其他没有耦合辅助信息的隐因子分解方法。例如,当 $K = 10$ 时,提出的模型相比 MF 的 RMSE 指标,在 MovieLens-1M 和 HetRec2011 数据集上分别提升 3.4% 和 3.1%。通过上述实验数据可以得出结论,采用本文的模型耦合物品的属性信息,有助于改善推荐质量;同时也证明所采用的 COS 度量指标确实能够在捕捉物品的相似性方面发挥作用。

另外,在 MovieLens-1M 和 HetRec2011 两个数据集上,当 $K = 40$ 时的 RMSE 和 MAE 值均高于 $K = 10$ 时的相应值。这说明矩阵分解仅仅需要少量的隐因子刻画用户和物品的特征。如果隐特征向量的维度过高,将带来一定程度的噪声,反而影响算法的推荐质量。

在上述实验中, HetRec2011 数据集在所有算法上面的推荐性能,无论从 RMSE 指标还是 MAE 指标来说,都要好于数据集 MovieLens-1M 上对应算法的性能。这主要是因为前者平均每个用户对物品的评分数量是后者的 2.4 倍,也就是后者的评分更为稀疏造成。

5.2.2 特征维度 K 的影响

为了研究特征维数对模型性能的影响,用不同的维度 K 进行测试,其结果如图 3 和 4 所示。在实验中,模型参数 β ,对于 MovieLens-1M 和 HetRec2011 数据集 β 分别设定为 0.2 和 0.3。 K 值由小变大,为 5~50,每次增长的步长为 5。

从实验结果可以看出,当维数 K 增长时, RMSE 和 MAE 的值先不断减小;然而当维数超过一定阈值时, RMSE 和 MAE 的值开始增长。其原因是:相对较大的维数可以更准确地刻画用户和物品的隐特征,从而具有更高的预测准确性;但当维数超过一定的阈值之后(例如,在 MovieLens-1M 中 K 为 15, HetRec2011 中 K 为 10),隐特征向量的维数太高反而引入噪声,使得模型的预测准确性降低。

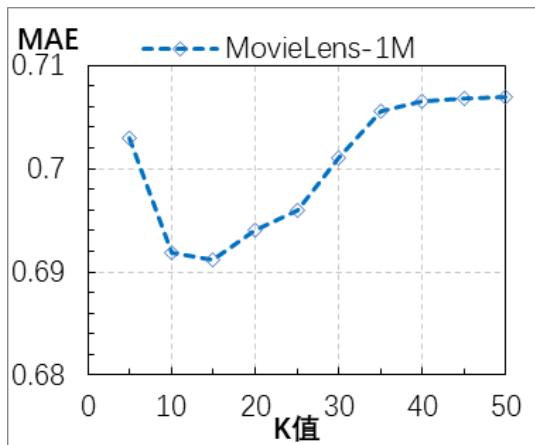


图3 维数 K 的影响(MovieLens-1M)

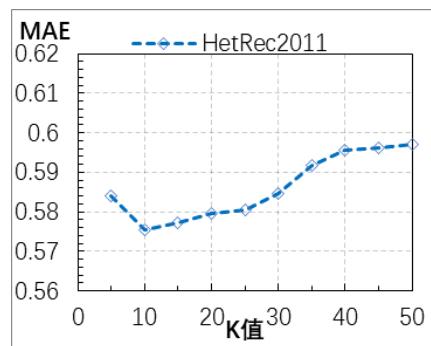


图4 维数 K 的影响(HetRec2011)

5.2.3 冷启动物品的推荐性能

推荐系统的主要挑战之一是很难把一个冷启动物品推荐给用户,因为用户对冷启动物品的评分非常少。为了验证我们方法在物品端冷启动问题上的有效性,对两个数据集按每个物品的评分数量,把物品分成 8 组,然后在 MF、NMF、PMF、以及本文的模型上进行对比实验。实验中的参数 K 定为 10;对 MovieLens-1M 和 HetRec2011 数据集 β 分别设定为 0.2 和 0.3;评价指标采用 MAE。

实验得出的预测评分结果如表 5 和 6 所示。从实验结果可以看出,在物品端冷启动问题上,本文的模型在各组物品分类中其预测准确性优于其他方法;特别是对于评分数较少的物品,本文的模型效果更为突出。例如,本文模型相比 MF 模型,在 MovieLens-1M 和 HetRec2011 数据集上的推荐性能分别提升 2.5% 和 4.3%。这说明本文模型能够有效缓解物品端冷启动。其主要原因是在矩阵分解过程中,采用了 COS 来捕捉物品之间的相似性,并用这种相似性来修正物品的隐特征向量。

表5 不同评分数量的物品组MAE值对比(MovieLens-1M)

序号	评分数	MF	NMF	PMF	Our Model
1	0	0.8567	0.8870	0.8673	0.7892
2	1~10	0.7914	0.8474	0.8264	0.7717
3	11~20	0.7698	0.8102	0.7705	0.7347
4	21~40	0.7332	0.7821	0.7518	0.7134
5	41~80	0.7138	0.7425	0.7227	0.6936
6	81~160	0.6983	0.72153	0.7029	0.6889
7	161~320	0.6946	0.7040	0.6983	0.6795
8	321~640	0.6842	0.6940	0.6866	0.6761

表6 不同评分数量的物品组MAE值对比(HetRec2011)

序号	评分数	MF	NMF	PMF	Our Model
1	0	0.7190	0.7295	0.7112	0.6809
2	1~10	0.6905	0.7125	0.6812	0.6607
3	11~20	0.6615	0.6819	0.6507	0.6418
4	21~40	0.6426	0.6515	0.6305	0.6201
5	41~80	0.6305	0.6410	0.6155	0.6005
6	81~160	0.6134	0.6175	0.6110	0.5795
7	161~320	0.6005	0.6050	0.5901	0.5754
8	321~640	0.5921	0.5945	0.5855	0.5705

根据以上的所有分析, 可以得出下列结论: a)在传统的矩阵分解协同过滤模型中耦合物品属性信息可以有效改善推荐算法的精度;b)COS 相似度能够准确地捕捉物品属性之间的相似性关系;c)本文提出的耦合物品信息的矩阵分解混合协同过滤模型能够有效地提高推荐精度并缓解物品冷启动。主要方法是通过耦合物品相似性来修正物品的隐特征向量, 使得具有较少评分数的物品的隐特征向量, 向相似属性的物品的隐特征向量逼近。

6 结束语

用户对物品的偏好不仅与用户与物品的交互评分相关, 用户及物品的相关辅助信息也是影响用户对物品喜好的重要原因。据此提出了一种新颖的耦合用户及物品辅助信息的矩阵分解混合协同过滤模型框架。该框架通过用户及物品辅助信息修正矩阵分解模型中的相关隐特征向量来实现。在文章的后半部分, 以此框架为基础, 提出了耦合物品相似度的矩阵分解混合协同过滤模型; 在模型中采用 COS 相似度来度量物品之间的相似度; 通过在 MovieLens-1M 和 hetrec2011 数据集上的实验证明, 此模型在提高推荐准确性以及缓解冷启动方面效果较为明显。

但是, 在矩阵分解模型中耦合物品相似度的方法, 很难提取非线性的特征表示。未来将使用机器学习技术(例如自编码器)针对物品的各种辅助信息来提取非线性特征, 然后再把它耦合到矩阵分解模型中。

参考文献:

- [1] Balabanović M, Shoham Y. Fab: content-based, collaborative recommendation [J]. Communications of the ACM, 1997, 40 (3): 66 - 72.
- [2] Deldjoo Y, Elahi M, Cremonesi P, et al. Content-based video recommendation system based on stylistic visual features [J]. Journal on Data Semantics, 2016, 5 (2): 99-113.
- [3] Shardanand U, Maes P. Social information filtering: algorithms for automating“word of mouth” [C]// Proc of Conference on Human Factors in Computing Systems. New York: ACM Press, 1995: 210-217.
- [4] Bell R M, Koren Y. Lessons from the netflix prize challenge [J]. ACM SIGKDD Explorations Newsletter, 2007, 9 (2): 75-79.
- [5] Koren Y. Factor in the neighbors: Scalable and accurate collaborative filtering [J]. ACM Trans on Knowledge Discovery from Data, 2010, 4 (1): 1-24.
- [6] Jian Wei, He Jianhua, Chen Kai, et al. Collaborative filtering and deep learning based recommendation system for cold start items [J]. Expert Systems with Applications, 2016, 69: 29-39.
- [7] Koren Y, Bell R. Advances in collaborative filtering. in: recommender systems handbook [M]. 2nd ed. [S. I.] : Springer, 2015: 145-186.
- [8] Agichtein E, Brill E, Dumais S. Improving web search ranking by incorporating user behavior information [C]// Proc of International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2006: 19-26.
- [9] Joachims T, Granka L, Pan B, et al. Accurately interpreting clickthrough data as implicit feedback [C]// Proc of International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2005: 154-161.
- [10] Hu Yifan, Koren Y, Volinsky C, et al. Collaborative filtering for implicit feedback datasets [C]// Proc of International Conference on Data Mining. Washington: IEEE Press, 2008: 263-272.
- [11] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions [J]. IEEE Trans on Knowledge & Data Engineering, 2005, 17 (6): 734-749.
- [12] Manzato M G. Discovering latent factors from movies genres for enhanced recommendation [C]// Proc of the 6th ACM Conference on Recommender Systems. New York: ACM Press, 2012: 249-252.
- [13] Manzato M G. gSVD+: supporting implicit feedback on recommender systems with metadata awareness [C]// Proc of ACM Symposium on Applied Computing. Coimbra: Association for Computing Machinery. New York: ACM Press, 2013: 908-913.
- [14] Gantner Z, Drumond L, Freudenthaler C, et al. Learning attribute-to-feature mappings for cold-start recommendations [C]// Proc of IEEE International Conference on Data Mining. [S. I.] : IEEE Press, 2010: 176-185.
- [15] Enrich M, Braunhofer M, Ricci F. Cold-start management with cross-domain collaborative filtering and tags [C]// Proc of International Conference on Electronic Commerce and Web Technologies. [S. I.] : Springer Press, 2013: 101-112.
- [16] Fernández-Tobías I, Cantador I. Exploiting social tags in matrix factorization models for cross-domain collaborative filtering [C]// Proc of the 1st Workshop on New Trends in Content-based Recommender Systems. New York: ACM Press, 2014: 34-41.
- [17] Ge Mouzhi, Elahi M, Ricci F, et al. Using tags and latent factors in a food recommender system [C]// Proc of International Conference on Digital Health. New York: ACM Press, 2015: 105-112.
- [18] Ma Hao, Yang Haixuan, Lyu M R, et al. SoRec: social recommendation using probabilistic matrix factorization [C]// Proc of Conference on Information and Knowledge Management. New York: ACM Press, 2008: 931-940.
- [19] Ma Hao, King I, Lyu M R. Learning to recommend with social trust ensemble [C]// Proc of International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2009: 203-210.
- [20] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks [C]// Proc of ACM Conference on Recommender Systems. New York: ACM Press, 2010: 135-142.
- [21] Ma Hao, Zhou Dengyong, Liu Chao, et al. Recommender systems with social regularization [C]// Proc of ACM International Conference on Web Search and Data Mining. New York: ACM Press 2011: 287-296.

- [22] Wang Hao, Wang Naiyan, and Yeung D Y. Collaborative deep learning for recommender systems [C]// Proc of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2015: 1235–1244.
- [23] Zhang Shuai, Yao Lina, Xu Xiwei. 2017. AutoSVD+: an efficient hybrid collaborative filtering model via contractive autoencoders [C]// Proc of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2017: 957-960.
- [24] Kannan R, Ishteva M, Drake B, et al. Bounded matrix low rank approximation [C]// Proc of the 12th IEEE International Conference on Data Mining, 2016: 319-328.
- [25] Deerwester S, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis [J]. Journal of the American Society for Information Science, 2010, 41 (6): 391-407.
- [26] Tikk D. Matrix factorization and neighbor based algorithms for the netflix prize problem [C]// Proc of ACM Conference on Recommender Systems. New York: ACM Press, 2008: 267-274.
- [27] Gan Guojun, Ma Chaoqun, Wu Jianhong. Data clustering: theory, algorithms and applications [M]. [S. I.] : Society for Industrial and Applied Mathematics, 2007: 44-51.
- [28] Wilson D R, Martinez T R. Improved heterogeneous distance functions [J]. Journal of Artificial Intelligence Research, 1997, 11 (1): 1-34.
- [29] Cost S, Salzberg S. A weighted nearest neighbor algorithm for learning with symbolic features [J]. Machine Learning, 1993, 10 (1): 57-78.
- [30] Boriah S, Chandola V, Kumar V, et al. Similarity measures for categorical data: a comparative evaluation [C]// Proc of SIAM International Conference on Data Mining. [S. I.] : SIAM, 2008: 243-254.
- [31] Cao Longbing, Ou Yuming, Yu P S. Coupled behavior analysis with applications [J]. IEEE Trans on Knowledge & Data Engineering, 2011, 24 (8): 1378-1392.
- [32] Wang Can, Cao Longbing, Wang Mingchun, et al. Coupled nominal similarity in unsupervised learning [C]// Proc of the 20th ACM International Conference on Information and Knowledge Management. New York: ACM Press, 2011: 973-978.
- [33] Wang Can, Dong Xiangjun, Zhou Fei, et al. Coupled attribute similarity learning on categorical data [J]. IEEE Trans on Neural Networks & Learning Systems, 2015, 26 (4): 781-797.
- [34] Freyne J, Berkovsky S. Evaluating recommender systems for supportive technologies. [M]// User Modeling And Adaptation For Daily Routines. London: Springer, 2013: 195-217.
- [35] Koren, Y, Bell, R, Volinsky, C. Matrix factorization techniques for recommender systems [J]. Computer, 2009, 42 (8): 30-37
- [36] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization [J]. Nature, 1999, 401 (6755): 788–791.
- [37] Lee D D, Seung H S. Algorithms for non-negative matrix factorization [C]// Proc of International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2000: 535-541.
- [38] Salakhutdinov R, Mnih A. Probabilistic matrix factorization [C]// Proc of International Conference on Neural Information Processing Systems. [S. I.] : Curran Associates Inc. , 2007: 1257-1264.
- [39] Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo [C]// Proc of International Conference on Machine Learning. New York: ACM Press, 2008: 880-887.